PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

solver process and data stores and data structures

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **solver process** and **data stores** and **data structures** Found **103,429** of **151,219**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Design and data structure of fully adaptive, multigrid, finite-element software

Maria-Cecilia Rivara

August 1984 **ACM Transactions on Mathematical Software (TOMS)**, Volume 10 Issue 3

Full text available: pdf(1.40 MB) Additional Information: full citation, references, citings, index terms, review

2 SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems

Xiaoye S. Li, James W. Demmel

June 2003 **ACM Transactions on Mathematical Software (TOMS)**, Volume 29 Issue 2

Full text available: pdf(659.03 KB) Additional Information: full citation, abstract, references, citings, index terms

We present the main algorithmic features in the software package SuperLU_DIST, a distributed-memory sparse direct solver for large sets of linear equations. We give in detail our parallelization strategies, with a focus on scalability issues, and demonstrate the software's parallel performance and scalability on current machines. The solver is based on sparse Gaussian elimination, with an innovative static pivoting strategy proposed earlier by the authors. The main advantage of static pivoting o ...

Keywords: Sparse direct solver, distributed-memory computers, parallelism, scalability, supernodal factorization

3 Automatic test data generation using constraint solving techniques

Arnaud Gotlieb, Bernard Botella, Michel Rueher

March 1998 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1998 ACM SIGSOFT international symposium on Software testing and analysis**, Volume 23 Issue 2

Full text available: pdf(841.83 KB) Additional Information: full citation, abstract, references, citings, index terms

Automatic test data generation leads to identify input values on which a selected point in a procedure is executed. This paper introduces a new method for this problem based on constraint solving techniques. First, we statically transform a procedure into a constraint system by using well-known "Static Single Assignment" form and control-dependencies. Second, we solve this system to check whether at least one feasible control flow path going through the selected point exists and to generate test ...

Keywords: automatic test data generation, constraint solving techniques, global constraints, structural testing

4 Type systems for distributed data structures



Ben Liblit, Alexander Aiken

January 2000 **Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Full text available: pdf(1.64 MB) Additional Information: full citation, abstract, references, citings, index terms

Distributed-memory programs are often written using a global address space: any process can name any memory location on any processor. Some languages completely hide the distinction between local and remote memory, simplifying the programming model at some performance cost. Other languages give the programmer more explicit control, offering better potential performance but sacrificing both soundness and ease of use. Through a series of progressively richer type systems, we formal ...

5 PYTHIA-II: a knowledge/database system for managing performance data and recommending scientific software



Elias N. Houstis, Ann C. Catlin, John R. Rice, Vassilios S. Verykios, Naren Ramakrishnan, Catherine E. Houstis

June 2000 **ACM Transactions on Mathematical Software (TOMS)**, Volume 26 Issue 2

Full text available: pdf(796.16 KB) Additional Information: full citation, abstract, references, citings, index terms

Often scientists need to locate appropriate software for their problems and then select from among many alternatives. We have previously proposed an approach for dealing with this task by processing performance data of the targeted software. This approach has been tested using a customized implementation referred to as PYTHIA. This experience made us realize the complexity of the algorithmic discovery of knowledge from performance data and of the management of these data together with the d ...

Keywords: data mining, inductive logic programming, knowledge discovery in databases, knowledge-based systems, performance evaluation, recommender systems, scientific software

6 Multi Relational Data Mining (MRDM): Scalability and efficiency in multi-relational data mining



Hendrik Blockeel, Michèle Sebag

July 2003 **ACM SIGKDD Explorations Newsletter**, Volume 5 Issue 1

Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings

Efficiency and Scalability have always been important concerns in the field of data mining, and are even more so in the multi-relational context, which is inherently more complex. The issue has been receiving an increasing amount of attention during the last few years, and quite a number of theoretical results, algorithms and implementations have been presented that explicitly aim at improving the efficiency and Scalability of multi-relational data mining approaches. With this article we attempt ...

7 Solving partial differential equations in a data-driven multiprocessor environment



J. L. Gaudiot, C. M. Lin, M. Hosseiniyar

May 1988 **ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture**, Volume 16 Issue 2

Full text available: pdf(856.94 KB) Additional Information: full citation, abstract, references, citings, index terms

Partial differential equations can be found in a host of engineering and scientific problems. The emergence of new parallel architectures has spurred research in the definition of parallel PDE solvers. Concurrently, highly programmable systems such as data-flow architectures have been proposed for the exploitation of large scale parallelism. The implementation of some Partial Differential Equation solvers (such as the Jacobi method) on a tagged token data-flow graph is demonstrated here. As ...

8 Supporting shared data structures on distributed memory architectures

C. Koelbel, P. Mehrotra, J. Van Rosendale

February 1990 **ACM SIGPLAN Notices , Proceedings of the second ACM SIGPLAN symposium on Principles & practice of parallel programming**, Volume 25 Issue 3

Full text available: pdf(1.14 MB) Additional Information: full citation, abstract, references, citings, index terms

Programming nonshared memory systems is more difficult than programming shared memory systems, since there is no support for shared data structures. Current programming languages for distributed memory architectures force the user to decompose all data structures into separate pieces, with each piece "owned" by one of the processors in the machine, and with all communication explicitly specified by low-level message-passing primitives. This paper presents a new programming envir ...

9 Climate data assimilation on a massively parallel Supercomputer

Hong Q. Ding, Robert D. Ferraro

November 1996 **Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(82.48 KB) Additional Information: full citation, abstract, references, citings, index terms

We have designed and implemented a set of highly efficient and highly scalable algorithms for an unstructured computational package, the PSAS data assimilation package, as demonstrated by detailed performance analysis of systematic runs on up to 512-nodes of an Intel Paragon. The preconditioned Conjugate Gradient solver achieves a sustained 18 Gflops performance. Consequently, we achieve an unprecedented 100-fold reduction in time to solution on the Intel Paragon over a single head of a Cra ...

10 A data-level parallel linear-quadratic penalty algorithm for multicommodity network flows

Mustafa Ç. Pinar, Stavros A. Zenios

December 1994 **ACM Transactions on Mathematical Software (TOMS)**, Volume 20 Issue 4

Full text available: pdf(1.31 MB) Additional Information: full citation, abstract, references, index terms

We describe the development of a data-level, massively parallel, software system for the solution of multicommodity network flow problems. Using a smooth linear-quadratic penalty (LQP) algorithm we transform the multicommodity network flow problem into a sequence of independent min-cost network flow subproblems. The solution of these problems is coordinated via a simple, dense, nonlinear master program to obtain a solution that is feasible within some user-specified tolerance to the origina ...

Keywords: massively parallel algorithms, multicommodity network problems, parallel optimization

11 Information structures: tools in problem solving

M. E. D'Imperio
December 1969 **ACM SIGMOD Record**, Volume 1 Issue 2
Full text available: pdf(1.56 MB) Additional Information: full citation, references

12 Information structures: tools in problem solving
M. E. D'Imperio
December 1969 **ACM SIGMOD Record**, Volume 1 Issue 2
Full text available: pdf(1.56 MB) Additional Information: full citation, references

13 Algorithm 817 P2MESH: generic object-oriented interface between 2-D unstructured meshes and FEM/FVM-based PDE solvers
Enrico Bertolazzi, Gianmarco Manzini
March 2002 **ACM Transactions on Mathematical Software (TOMS)**, Volume 28 Issue 1
Full text available: pdf(259.04 KB) Additional Information: full citation, abstract, references, citings, index terms

The software interface P2MESH is a collection of C++ class templates suitable for developing prototypes of high-performance PDE solvers on unstructured 2-D meshes. P2MESH supports several discretization methods on triangles and quadrilaterals, such as finite volume or finite element. The design philosophy of P2MESH does not consider specific model problems or built-in approximation algorithms. The software package is general purpose and it may also be used as a building block in the implementati ...

Keywords: Finite Element, Finite Volume, Object-Oriented programming, PDE solvers, unstructured mesh

14 An introductory course in data structures with applications
J. P. Tremblay, P. G. Sorenson
September 1975 **ACM SIGCSE Bulletin**, Volume 7 Issue 3
Full text available: pdf(614.81 KB) Additional Information: full citation, abstract, citings, index terms

This paper describes a two semester introductory course in data (information) structures for the undergraduate computer science student that has evolved at the University of Saskatchewan, Saskatoon. The philosophy and organization of such a course are discussed. A comparison is made between the course described and data structure courses proposed by two commitees 'on curricula.

15 Programming data parallel algorithms on distributed memory using Kali
Charles Koelbel, Piyush Mehrotra
June 1991 **Proceedings of the 5th international conference on Supercomputing**
Full text available: pdf(937.29 KB) Additional Information: full citation, references, citings, index terms

16 The design, implementation, and evaluation of a symmetric banded linear solver for distributed-memory parallel computers
Anshul Gupta, Fred G. Gustavson, Mahesh Joshi, Sivan Toledo
March 1998 **ACM Transactions on Mathematical Software (TOMS)**, Volume 24 Issue 1
Full text available: pdf(213.58 KB) Additional Information: full citation, abstract, references, citings, index terms, review

This article describes the design, implementation, and evaluation of a parallel algorithm for the Cholesky factorization of symmetric banded matrices. The algorithm is part of IBM's parallel engineering and scientific subroutine library version 1.2 and is compatible with ScaLAPACK's banded solver. Analysis, as well as experiments on an IBM SP2 distributed-memory parallel computer, shows that the algorithm efficiently factors banded matrices with wide bandwidth. For example, a 31-mode SP2 fa ...

Keywords: Banded matrices, Cholesky factorization, distributed memory, parallel algorithm

17 Compression of particle data from hierarchical approximate methods



Dow-Yung Yang, Ananth Grama, Vivek Sarin, Naren Ramakrishnan
September 2001 **ACM Transactions on Mathematical Software (TOMS)**, Volume 27 Issue 3

Full text available: pdf(614.22 KB) Additional Information: full citation, abstract, references, index terms

This article presents an analytical and computational framework for the compression of particle data resulting from hierarchical approximate treecodes such as the *Barnes--Hut* and *Fast Multipole Methods*. Due to approximations introduced by hierarchical methods, various parameters (such as position, velocity, acceleration, potential) associated with a particle can be bounded by distortion radii. Using this distortion radii, we develop storage schemes that guarantee error bounds while ...

Keywords: Astrophysics, Barnes--Hut, Fast Multipole Method, data compression and analysis, materials simulation, molecular dynamics, particle dynamics

18 A parallel direct solver for large sparse highly unsymmetric linear systems



Iain S. Duff, Jennifer A. Scott
June 2004 **ACM Transactions on Mathematical Software (TOMS)**, Volume 30 Issue 2

Full text available: pdf(232.43 KB) Additional Information: full citation, abstract, references, index terms, review

The need to solve large sparse linear systems of equations efficiently lies at the heart of many applications in computational science and engineering. For very large systems when using direct factorization methods of solution, it can be beneficial and sometimes necessary to use multiple processors, because of increased memory availability as well as reduced factorization time. We report on the development of a new parallel code that is designed to solve linear systems with a highly unsymmetric ...

Keywords: Gaussian elimination, Sparse matrices, highly unsymmetric linear systems, parallel processing

19 Applications and problem solving environments: *Roccom*: an object-oriented, data-centric software integration framework for multiphysics simulations



Xiangmin Jiao, Michael T. Campbell, Michael T. Heath
June 2003 **Proceedings of the 17th annual international conference on Supercomputing**

Full text available: pdf(265.82 KB) Additional Information: full citation, abstract, references, index terms

We describe an object-oriented software integration framework, *Roccom*, abstracted from our five years of experience in developing a complex, integrated code for rocket simulation. *Roccom* provides a flexible mechanism for inter-module data exchange and function invocation in parallel multiphysics simulations. It is designed to minimize user effort and code changes for integration, facilitate interoperability between different programming languages (in particular, C++ and Fortran 90) ...

Keywords: middleware, multiphysics simulation, object-oriented design, problem solving environments, system integration

20 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4

Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

solver process and data stores and data structures and assignr

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
solver process and **data stores** and **data structures** and **assignment** and **first metric**

Found **98,990** of **151,219**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Efficient load balancing and data remapping for adaptive grid calculations

Leonid Oliker, Rupak Biswas

June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures**

Full text available: pdf(1.30 MB) Additional Information: full citation, references, citings, index terms

2 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4

Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

3 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

4 HPFBench: a high performance Fortran benchmark suite

Y. Charlie Hu, Guohua Jin, S. Lennart Johnsson, Dimitris Kehagias, Nadia Shalaby
March 2000 **ACM Transactions on Mathematical Software (TOMS)**, Volume 26 Issue 1

Full text available: pdf(274.52 KB) Additional Information: full citation, abstract, references, index terms

The high performance Fortran (HPF) benchmark suite HPFBench is designed for evaluating the HPF language and compilers on scalable architectures. The functionality of the benchmarks covers scientific software library functions and application kernels that reflect the computational structure and communication patterns in fluid dynamic simulations, fundamental physics, and molecular studies in chemistry and biology. The benchmarks are characterized in terms of FLOP count, memory usage, communi ...

Keywords: benchmarks, compilers, high performance Fortran

5 Experience Using Multiprocessor Systems—A Status Report

Anita K. Jones, Peter Schwarz
June 1980 **ACM Computing Surveys (CSUR)**, Volume 12 Issue 2

Full text available: pdf(4.46 MB) Additional Information: full citation, references, citings, index terms

6 Mapping performance data for high-level and data views of parallel program performance

R. Bruce Irvin, Barton P. Miller
January 1996 **Proceedings of the 10th international conference on Supercomputing**

Full text available: pdf(994.50 KB) Additional Information: full citation, references, citings, index terms

7 Automatic generation of intelligent diagram editors

Sitt Sen Chok, Kim Marriott
September 2003 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 10 Issue 3

Full text available: pdf(1.43 MB) Additional Information: full citation, abstract, references, index terms

The *intelligent diagram* is a recent metaphor for diagramming in which the underlying graphic editor parses the diagram as it is being constructed, performing error correction and collecting geometric constraints that capture the relationships between diagram components. During diagram manipulation a constraint solver uses these geometric constraints to maintain the diagram's semantics. We introduce the Penguins system. This automates the development of graphical editors that support the i ...

Keywords: Constraint multi-set grammars, constraint solving, diagram interaction, diagram parsing, intelligent diagram, pen-based computing

8 The role of time in information processing: a survey

A. Bolour, T. L. Anderson, L. J. Dekeyser, H. K. T. Wong
April 1982 **ACM SIGMOD Record**, Volume 12 Issue 3

Full text available: pdf(2.16 MB) Additional Information: full citation, references, citings

9

Programming languages for distributed computing systems
Henri E. Bal, Jennifer G. Steiner, Andrew S. Tanenbaum
September 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 3

Full text available: pdf(6.50 MB) Additional Information: full citation, abstract, references, citings, index terms, review

When distributed systems first appeared, they were programmed in traditional sequential languages, usually with the addition of a few library procedures for sending and receiving messages. As distributed applications became more commonplace and more sophisticated, this ad hoc approach became less satisfactory. Researchers all over the world began designing new programming languages specifically for implementing distributed applications. These languages and their history, their underlying pr ...

10 A framework for call graph construction algorithms
David Grove, Craig Chambers
November 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 6

Full text available: pdf(1.36 MB) Additional Information: full citation, abstract, references, citings, index terms

A large number of call graph construction algorithms for object-oriented and functional languages have been proposed, each embodying different tradeoffs between analysis cost and call graph precision. In this article we present a unifying framework for understanding call graph construction algorithms and an empirical comparison of a representative set of algorithms. We first present a general parameterized algorithm that encompasses many well-known and novel call graph construction algorithms. W ...

Keywords: Call graph construction, control flow analysis, interprocedural analysis

11 Technical reports
SIGACT News Staff
January 1980 **ACM SIGACT News**, Volume 12 Issue 1

Full text available: pdf(5.28 MB) Additional Information: full citation

12 Software unit test coverage and adequacy
Hong Zhu, Patrick A. V. Hall, John H. R. May
December 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 4

Full text available: pdf(477.42 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Objective measurement of test quality is one of the key issues in software testing. It has been a major research focus for the last two decades. Many test criteria have been proposed and studied for this purpose. Various kinds of rationales have been presented in support of one criterion or another. We survey the research work in this area. The notion of adequacy criteria is examined together with its role in software dynamic testing. A review of criteria classification is followed by a sum ...

Keywords: comparing testing effectiveness, fault detection, software unit test, test adequacy criteria, test coverage, testing methods

13 Storage: An evaluation of multi-resolution storage for sensor networks
Deepak Ganesan, Ben Greenstein, Denis Perelyubskiy, Deborah Estrin, John Heidemann
November 2003 **Proceedings of the 1st international conference on Embedded**

networked sensor systems

Full text available: pdf(299.34 KB) Additional Information: full citation, abstract, references, citings

Wireless sensor networks enable dense sensing of the environment, offering unprecedented opportunities for observing the physical world. Centralized data collection and analysis adversely impact sensor node lifetime. Previous sensor network research has, therefore, focused on in network aggregation and query processing, but has done so for applications where the features of interest are known a priori. When features are not known a priori, as is the case with many scientific applications in dens ...

14 A comparison of three programming models for adaptive applications on the Origin2000

Hongzhang Shan, Jaswinder P. Singh, Leonid Oliker, Rupak Biswas

November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(239.30 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Adaptive applications have computational workloads and communication patterns which change unpredictably at runtime, requiring load balancing to achieve scalable performance on parallel machines. Efficient parallel implementation of such adaptive application is therefore a challenging task. In this paper, we compare the performance of and the programming effort required for two major classes of adaptive applications under three leading parallel programming models on an SGI Origin 2000 syste ...

15 Mechanical program analysis

Ben Wegbreit

September 1975 **Communications of the ACM**, Volume 18 Issue 9

Full text available: pdf(1.29 MB) Additional Information: full citation, abstract, references, citings, index terms

One means of analyzing program performance is by deriving closed-form expressions for their execution behavior. This paper discusses the mechanization of such analysis, and describes a system, Metric, which is able to analyze simple Lisp programs and produce, for example, closed-form expressions for their running time expressed in terms of size of input. This paper presents the reasons for mechanizing program analysis, describes the operation of Metric, explains its implementation, and disc ...

Keywords: Lisp, algebraic manipulation, analysis of algorithms, analysis of programs, difference equations, execution behavior, execution time, generating functions, list processing, performance analysis, programming languages

16 Data placement in Bubba

George Copeland, William Alexander, Ellen Boughter, Tom Keller

June 1988 **ACM SIGMOD Record , Proceedings of the 1988 ACM SIGMOD international conference on Management of data**, Volume 17 Issue 3

Full text available: pdf(1.41 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper examines the problem of data placement in Bubba, a highly-parallel system for data-intensive applications being developed at MCC. "Highly-parallel" implies that load balancing is a critical performance issue. "Data-intensive" means data is so large that operations should be executed where the data resides. As a result, data placement becomes a critical performance issue. In general, determining the optimal placement of d ...

17

Scalable analysis techniques for microprocessor performance counter metrics
Dong H. Ahn, Jeffrey S. Vetter
November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, index terms

Contemporary microprocessors provide a rich set of integrated performance counters that allow application developers and system architects alike the opportunity to gather important information about workload behaviors. Current techniques for analyzing data produced from these counters use raw counts, ratios, and visualization techniques help users make decisions about their application performance. While these techniques are appropriate for analyzing data from one process, they do not scale easi ...

18 Analysis and comparison of two general sparse solvers for distributed memory computers
Patrick R. Amestoy, Iain S. Duff, Jean-Yves L'excellent, Xiaoye S. Li
December 2001 **ACM Transactions on Mathematical Software (TOMS)**, Volume 27 Issue 4

Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper provides a comprehensive study and comparison of two state-of-the-art direct solvers for large sparse sets of linear equations on large-scale distributed-memory computers. One is a multifrontal solver called MUMPS, the other is a supernodal solver called superLU. We describe the main algorithmic features of the two solvers and compare their performance characteristics with respect to uniprocessor speed, interprocessor communication, and memory requirements. For both solvers, preorderi ...

Keywords: Sparse direct solvers, distributed-memory computers, multifrontal and supernodal factorizations, parallelism

19 A Status Report on Computing Algorithms for Mathematical Programming
William W. White
September 1973 **ACM Computing Surveys (CSUR)**, Volume 5 Issue 3

Full text available: pdf(3.02 MB) Additional Information: full citation, references, citings, index terms

20 The design and implementation of an intentional naming system
William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, Jeremy Lilley
December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles**, Volume 33 Issue 5

Full text available: pdf(1.77 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents the design and implementation of the Intentional Naming System (INS), a resource discovery and service location system for dynamic and mobile networks of devices and computers. Such environments require a naming system that is (i) expressive, to describe and make requests based on specific properties of services, (ii) responsive, to track changes due to mobility and performance, (iii) robust, to handle failures, and (iv) easily configurable. INS uses a simple language based o ...

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

solver process and data stores and first metric and second met

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
solver process and **data stores** and **first metric** and **second metric**

Found **102,185** of **151,219**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

2 Progress-based regulation of low-importance processes

John R. Douceur, William J. Bolosky

December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles**, Volume 33 Issue 5

Full text available: pdf(1.53 MB) Additional Information: full citation, abstract, references, citings, index terms

MS Manners is a mechanism that employs progress-based regulation to prevent resource contention with low-importance processes from degrading the performance of high-importance processes. The mechanism assumes that resource contention that degrades the performance of a high-importance process will also retard the progress of the low-importance process. MS Manners detects this contention by monitoring the progress of the low-importance process and inferring resource contention from a drop in the p ...

Keywords: process priority, progress-based feedback, symmetric resource contention

3 HPFBench: a high performance Fortran benchmark suite

Y. Charlie Hu, Guohua Jin, S. Lennart Johnsson, Dimitris Kehagias, Nadia Shalaby

March 2000 **ACM Transactions on Mathematical Software (TOMS)**, Volume 26 Issue 1

Full text available: pdf(274.52 KB) Additional Information: full citation, abstract, references, index terms

The high performance Fortran (HPF) benchmark suite HPFBench is designed for evaluating the HPF language and compilers on scalable architectures. The functionality of the

benchmarks covers scientific software library functions and application kernels that reflect the computational structure and communication patterns in fluid dynamic simulations, fundamental physics, and molecular studies in chemistry and biology. The benchmarks are characterized in terms of FLOP count, memory usage, communi ...

Keywords: benchmarks, compilers, high performance Fortran

4 Scalable analysis techniques for microprocessor performance counter metrics

Dong H. Ahn, Jeffrey S. Vetter

November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, index terms

Contemporary microprocessors provide a rich set of integrated performance counters that allow application developers and system architects alike the opportunity to gather important information about workload behaviors. Current techniques for analyzing data produced from these counters use raw counts, ratios, and visualization techniques help users make decisions about their application performance. While these techniques are appropriate for analyzing data from one process, they do not scale easi ...

5 Minerva: An automated resource provisioning tool for large-scale storage systems

Guillermo A. Alvarez, Elizabeth Borowsky, Susie Go, Theodore H. Romer, Ralph Becker-Szendy, Richard Golding, Arif Merchant, Mirjana Spasojevic, Alistair Veitch, John Wilkes

November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Full text available: pdf(701.98 KB) Additional Information: full citation, abstract, references, index terms

Enterprise-scale storage systems, which can contain hundreds of host computers and storage devices and up to tens of thousands of disks and logical volumes, are difficult to design. The volume of choices that need to be made is massive, and many choices have unforeseen interactions. Storage system design is tedious and complicated to do by hand, usually leading to solutions that are grossly over-provisioned, substantially under-performing or, in the worst case, both.To solve the configuration ni ...

Keywords: Disk array, RAID, automatic design

6 Probabilistic wavelet synopses

Minos Garofalakis, Phillip B. Gibbons

March 2004 **ACM Transactions on Database Systems (TODS)**, Volume 29 Issue 1

Full text available: pdf(668.62 KB) Additional Information: full citation, abstract, references, index terms

Recent work has demonstrated the effectiveness of the wavelet decomposition in reducing large amounts of data to compact sets of wavelet coefficients (termed "wavelet synopses") that can be used to provide fast and reasonably accurate approximate query answers. A major shortcoming of these existing wavelet techniques is that the quality of the approximate answers they provide varies widely, even for identical queries on nearly identical values in distinct parts of the data. As a result, users ha ...

Keywords: Wavelets, approximate query processing, data synopses, randomized rounding

7 Efficient load balancing and data remapping for adaptive grid calculations

Leonid Oliker, Rupak Biswas

June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures**

Full text available: pdf(1.30 MB) Additional Information: full citation, references, citings, index terms

8 Parallel execution of prolog programs: a survey
Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo
July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4
Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

9 Performance monitoring: METRIC: tracking down inefficiencies in the memory hierarchy via binary rewriting
Jaydeep Marathe, Frank Mueller, Tushar Mohan, Bronis R. de Supinski, Sally A. McKee, Andy Yoo
March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**
Full text available: pdf(1.77 MB) Publisher Site Additional Information: full citation, abstract, references, index terms

In this paper, we present METRIC, an environment for determining memory inefficiencies by examining data traces. METRIC is designed to alter the performance behavior of applications that are mostly constrained by their latency to resolve memory references. We make four primary contributions in this paper. First, we present methods to extract partial data traces from running applications by observing their memory behavior via dynamic binary rewriting. Second, we present a methodology to represent ...

10 Software process validation: quantitatively measuring the correspondence of a process to a model
Jonathan E. Cook, Alexander L. Wolf
April 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 8 Issue 2
Full text available: pdf(267.69 KB) Additional Information: full citation, abstract, references, citings, index terms

To a great extent, the usefulness of a formal model of a software process lies in its ability to accurately predict the behavior of the executing process. Similarly, the usefulness of an executing process lies largely in its ability to fulfill the requirements embodied in a formal model of the process. When process models and process executions diverge, something significant is happening. We have developed techniques for uncovering and measuring the discrepancies between models and executio ...

Keywords: balboa, process validation, software process, tools

11 Performance assertion checking
Sharon E. Perl, William E. Weihl

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5

Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, citings, index terms

Performance assertion checking is an approach to automating the testing of performance properties of complex systems. System designers write assertions that capture expectations for performance; these assertions are checked automatically against monitoring data to detect potential performance bugs. Automatically checking expectations allows a designer to test a wide range of performance properties as a system evolves: data that meets expectations can be discarded automatically, focusing a ...

12 Stream query processing I: Approximate join processing over data streams

Abhinandan Das, Johannes Gehrke, Mirek Riedewald

June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on Management of data**

Full text available: pdf(282.87 KB) Additional Information: full citation, abstract, references, citings, index terms

We consider the problem of approximating sliding window joins over data streams in a data stream processing system with limited resources. In our model, we deal with resource constraints by shedding load in the form of dropping tuples from the data streams. We first discuss alternate architectural models for data stream join processing, and we survey suitable measures for the quality of an approximation of a set-valued query result. We then consider the number of generated result tuples as the q ...

13 SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems

Xiaoye S. Li, James W. Demmel

June 2003 **ACM Transactions on Mathematical Software (TOMS)**, Volume 29 Issue 2

Full text available: pdf(659.03 KB) Additional Information: full citation, abstract, references, citings, index terms

We present the main algorithmic features in the software package SuperLU_DIST, a distributed-memory sparse direct solver for large sets of linear equations. We give in detail our parallelization strategies, with a focus on scalability issues, and demonstrate the software's parallel performance and scalability on current machines. The solver is based on sparse Gaussian elimination, with an innovative static pivoting strategy proposed earlier by the authors. The main advantage of static pivoting o ...

Keywords: Sparse direct solver, distributed-memory computers, parallelism, scalability, supernodal factorization

14 The design and implementation of an intentional naming system

William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, Jeremy Lilley

December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles**, Volume 33 Issue 5

Full text available: pdf(1.77 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents the design and implementation of the Intentional Naming System (INS), a resource discovery and service location system for dynamic and mobile networks of devices and computers. Such environments require a naming system that is (i) expressive, to describe and make requests based on specific properties of services, (ii) responsive, to track changes due to mobility and performance, (iii) robust, to handle failures, and (iv)

easily configurable. INS uses a simple language based o ...

15 Research sessions: compression: Wavelet synopses with error guarantees
Minos Garofalakis, Phillip B. Gibbons
June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**
Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, citings, index terms

Recent work has demonstrated the effectiveness of the wavelet decomposition in reducing large amounts of data to compact sets of wavelet coefficients (termed "wavelet synopses") that can be used to provide fast and reasonably accurate approximate answers to queries. A major criticism of such techniques is that unlike, for example, random sampling, conventional wavelet synopses do not provide informative error guarantees on the accuracy of individual approximate answers. In fact, as this paper de ...

16 The design and implementation of SOLAR, a portable library for scalable out-of-core linear algebra computations
Sivan Toledo, Fred G. Gustavson
May 1996 **Proceedings of the fourth workshop on I/O in parallel and distributed systems: part of the federated computing research conference**
Full text available: pdf(1.63 MB) Additional Information: full citation, references, citings, index terms

17 Converting thread-level parallelism to instruction-level parallelism via simultaneous multithreading
Jack L. Lo, Joel S. Emer, Henry M. Levy, Rebecca L. Stamm, Dean M. Tullsen, S. J. Eggers
August 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 3
Full text available: pdf(526.39 KB) Additional Information: full citation, abstract, references, citings, index terms, review

To achieve high performance, contemporary computer systems rely on two forms of parallelism: instruction-level parallelism (ILP) and thread-level parallelism (TLP). Wide-issue super-scalar processors exploit ILP by executing multiple instructions from a single program in a single cycle. Multiprocessors (MP) exploit TLP by executing different threads in parallel on different processors. Unfortunately, both parallel processing styles statically partition processor resources, thus preventing t ...

Keywords: cache interference, instruction-level parallelism, multiprocessors, multithreading, simultaneous multithreading, thread-level parallelism

18 Mechanical program analysis
Ben Wegbreit
September 1975 **Communications of the ACM**, Volume 18 Issue 9
Full text available: pdf(1.29 MB) Additional Information: full citation, abstract, references, citings, index terms

One means of analyzing program performance is by deriving closed-form expressions for their execution behavior. This paper discusses the mechanization of such analysis, and describes a system, Metric, which is able to analyze simple Lisp programs and produce, for example, closed-form expressions for their running time expressed in terms of size of input. This paper presents the reasons for mechanizing program analysis, describes the operation of Metric, explains its implementation, and disc ...

Keywords: Lisp, algebraic manipulation, analysis of algorithms, analysis of programs,

difference equations, execution behavior, execution time, generating functions, list processing, performance analysis, programming languages

19 Information extraction as a basis for high-precision text classification

Ellen Riloff, Wendy Lehnert

July 1994 **ACM Transactions on Information Systems (TOIS)**, Volume 12 Issue 3

Full text available: pdf(2.79 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe an approach to text classification that represents a compromise between traditional word-based techniques and in-depth natural language processing. Our approach uses a natural language processing task called "information extraction" as a basis for high-precision text classification. We present three algorithms that use varying amounts of extracted information to classify texts. The relevancy signatures algorithm uses linguistic phrases; the a ...

Keywords: information extraction, text classification

20 Automatic generation of intelligent diagram editors

Sitt Sen Chok, Kim Marriott

September 2003 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 10 Issue 3

Full text available: pdf(1.43 MB) Additional Information: full citation, abstract, references, index terms

The *intelligent diagram* is a recent metaphor for diagramming in which the underlying graphic editor parses the diagram as it is being constructed, performing error correction and collecting geometric constraints that capture the relationships between diagram components. During diagram manipulation a constraint solver uses these geometric constraints to maintain the diagram's semantics. We introduce the Penguins system. This automates the development of graphical editors that support the i ...

Keywords: Constraint multi-set grammars, constraint solving, diagram interaction, diagram parsing, intelligent diagram, pen-based computing

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

solver process and data stores and assignment and devices

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
solver process and **data stores** and **assignment** and **devices**

Found **89,079** of **151,219**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Minerva: An automated resource provisioning tool for large-scale storage systems
Guillermo A. Alvarez, Elizabeth Borowsky, Susie Go, Theodore H. Romer, Ralph Becker-Szendy, Richard Golding, Arif Merchant, Mirjana Spasojevic, Alistair Veitch, John Wilkes
November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4
Full text available: pdf(701.98 KB) Additional Information: full citation, abstract, references, index terms

Enterprise-scale storage systems, which can contain hundreds of host computers and storage devices and up to tens of thousands of disks and logical volumes, are difficult to design. The volume of choices that need to be made is massive, and many choices have unforeseen interactions. Storage system design is tedious and complicated to do by hand, usually leading to solutions that are grossly over-provisioned, substantially under-performing or, in the worst case, both.To solve the configuration ni ...

Keywords: Disk array, RAID, automatic design

2 Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science
William F. Atchison, Samuel D. Conte, John W. Hamblen, Thomas E. Hull, Thomas A. Keenan, William B. Kehl, Edward J. McCluskey, Silvio O. Navarro, Werner C. Rheinboldt, Earl J. Schweppe, William Viavant, David M. Young
March 1968 **Communications of the ACM**, Volume 11 Issue 3
Full text available: pdf(6.63 MB) Additional Information: full citation, references, citings

Keywords: computer science academic programs, computer science bibliographies, computer science courses, computer science curriculum, computer science education, computer science graduate programs, computer science undergraduate programs

3 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

4 Experience Using Multiprocessor Systems—A Status Report
Anita K. Jones, Peter Schwarz
June 1980 **ACM Computing Surveys (CSUR)**, Volume 12 Issue 2
Full text available: pdf(4.48 MB) Additional Information: full citation, references, citings, index terms

5 A constructive algorithm for memory-aware task assignment and scheduling
Radoslaw Szymanek, Krzysztof Kuchcinski
April 2001 **Proceedings of the ninth international symposium on Hardware/software codesign**
Full text available: pdf(618.45 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a constructive algorithm for memory-aware task assignment and scheduling, which is a part of the prototype system MATAS. The algorithm is well suited for image and video processing applications which have hard memory constraints as well as constraints on cost, execution time, and resource usage. Our algorithm takes into account code and data memory constraints together with the other constraints. It can create pipelined implementations. The algorithm finds a task assignmen ...

Keywords: constraint programming, memory constraints, task assignment, task scheduling

6 Information structures: tools in problem solving
M. E. D'Imperio
December 1969 **ACM SIGMOD Record**, Volume 1 Issue 2
Full text available: pdf(1.56 MB) Additional Information: full citation, references

7 Information structures: tools in problem solving
M. E. D'Imperio
December 1969 **ACM SIGMOD Record**, Volume 1 Issue 2
Full text available: pdf(1.56 MB) Additional Information: full citation, references

8 A parallel direct solver for large sparse highly unsymmetric linear systems
Iain S. Duff, Jennifer A. Scott
June 2004 **ACM Transactions on Mathematical Software (TOMS)**, Volume 30 Issue 2
Full text available: pdf(232.43 KB) Additional Information: full citation, abstract, references, index terms, review

The need to solve large sparse linear systems of equations efficiently lies at the heart of many applications in computational science and engineering. For very large systems when using direct factorization methods of solution, it can be beneficial and sometimes necessary to use multiple processors, because of increased memory availability as well as reduced

factorization time. We report on the development of a new parallel code that is designed to solve linear systems with a highly unsymmetric ...

Keywords: Gaussian elimination, Sparse matrices, highly unsymmetric linear systems, parallel processing

9 A Survey of Interactive Graphical Systems for Mathematics
Lyle B. Smith
December 1970 **ACM Computing Surveys (CSUR)**, Volume 2 Issue 4
Full text available: pdf(5.05 MB) Additional Information: full citation, references, citings, index terms

10 Power minimization in IC design: principles and applications
Massoud Pedram
January 1996 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 1 Issue 1
Full text available: pdf(550.02 KB) Additional Information: full citation, abstract, references, citings, index terms

Low power has emerged as a principal theme in today's electronics industry. The need for low power has caused a major paradigm shift in which power dissipation is as important as performance and area. This article presents an in-depth survey of CAD methodologies and techniques for designing low power digital CMOS circuits and systems and describes the many issues facing designers at architectural, logical, and physical levels of design abstraction. It reviews some of the techniques and tool ...

Keywords: CMOS circuits, adiabatic circuits, computer-aided design of VLSI, dynamic power dissipation, energy-delay product, gated clocks, layout, low power layout, low power synthesis, lower-power design, power analysis and estimation, power management, power minimization and management, probabilistic analysis, silicon-on-insulator technology, statistical sampling, switched capacitance, switching activity, symbolic simulation, synthesis, system design

11 Taming the IXP network processor
Lal George, Matthias Blume
May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5
Full text available: pdf(159.27 KB) Additional Information: full citation, abstract, references, citings, index terms, review

We compile Nova, a new language designed for writing network processing applications, using a back end based on integer-linear programming (ILP) for register allocation, optimal bank assignment, and spills. The compiler's optimizer employs CPS as its intermediate representation; some of the invariants that this IR guarantees are essential for the formulation of a practical ILP model.Appel and George used a similar ILP-based technique for the IA32 to decide which variables reside in registers but ...

Keywords: Intel IXA, bank assignment, code generation, integer linear programming, network processors, programming languages, register allocation

12 Distributed operating systems
Andrew S. Tanenbaum, Robbert Van Renesse
December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4

Full text available: pdf(5.49 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

13 Computing as a discipline

D. E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, Paul R. Young

February 1989 **Communications of the ACM**, Volume 32 Issue 1

Full text available: pdf(1.68 MB) Additional Information: full citation, abstract, references, citings, index terms

The final report of the Task Force on the Core of Computer Science presents a new intellectual framework for the discipline of computing and a new basis for computing curricula. This report has been endorsed and approved for release by the ACM Education Board.

14 Storage: An evaluation of multi-resolution storage for sensor networks

Deepak Ganesan, Ben Greenstein, Denis Perelyubskiy, Deborah Estrin, John Heidemann

November 2003 **Proceedings of the 1st international conference on Embedded networked sensor systems**

Full text available: pdf(299.34 KB) Additional Information: full citation, abstract, references, citings

Wireless sensor networks enable dense sensing of the environment, offering unprecedented opportunities for observing the physical world. Centralized data collection and analysis adversely impact sensor node lifetime. Previous sensor network research has, therefore, focused on in network aggregation and query processing, but has done so for applications where the features of interest are known a priori. When features are not known a priori, as is the case with many scientific applications in dens ...

15 Balancing performance and flexibility with hardware support for network architectures

Ilija Hadžić, Jonathan M. Smith

November 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 4

Full text available: pdf(719.03 KB) Additional Information: full citation, abstract, references, index terms

The goals of performance and flexibility are often at odds in the design of network systems. The tension is common enough to justify an architectural solution, rather than a set of context-specific solutions. The Programmable Protocol Processing Pipeline (P4) design uses programmable hardware to selectively accelerate protocol processing functions. A set of field-programmable gate arrays (FPGAs) and an associated library of network processing modules implemented in hardware are augmented with so ...

Keywords: FPGA, P4, computer networking, flexibility, hardware, performance, programmable logic devices, programmable networks, protocol processing

16 A methodology for testing spreadsheets

Gregg Rothermel, Margaret Burnett, Lixin Li, Christopher Dupuis, Andrei Sheretov

January 2001 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 10 Issue 1

Full text available: pdf(353.65 KB) Additional Information: full citation, abstract, references, citings, index terms

Spreadsheet languages, which include commercial spreadsheets and various research systems, have had a substantial impact on end-user computing. Research shows, however, that spreadsheets often contain faults; thus, we would like to provide at least some of the benefits of formal testing methodologies to the creators of spreadsheets. This article presents a testing methodology that adapts data flow adequacy criteria and coverage monitoring to the task of testing spreadsheets. To accommodate ...

Keywords: software testing, spreadsheets

17 Run-time performance optimization of an FPGA-based deduction engine for SAT solvers

Andreas Dandalis, Viktor K. Prasanna

October 2002 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 7 Issue 4

Full text available: pdf(375.20 KB) Additional Information: full citation, abstract, references, citings, index terms

FPGAs are a promising technology for accelerating SAT solvers. Besides their high density, fine granularity, and massive parallelism, FPGAs provide the opportunity for run-time customization of the hardware based on the given SAT instance. In this article, a parallel deduction engine is proposed for backtrack search algorithms. The performance of the deduction engine is critical to the overall performance of the algorithm because, for any moderate SAT instance, millions of implications are deriv ...

Keywords: Adaptive computing, Boolean satisfiability, configurable, high performance, performance trade-offs, reconfigurable components, reconfigurable computing, reconfigurable systems

18 Technical reports

SIGACT News Staff

January 1980 **ACM SIGACT News**, Volume 12 Issue 1

Full text available: pdf(5.28 MB) Additional Information: full citation

19 Strategic directions in constraint programming

Pascal Van Hentenryck, Vijay Saraswat

December 1996 **ACM Computing Surveys (CSUR)**, Volume 28 Issue 4

Full text available: pdf(402.06 KB) Additional Information: full citation, references, citings, index terms

20 PSBLAS: a library for parallel linear algebra computation on sparse matrices

Salvatore Filippone, Michele Colajanni

December 2000 **ACM Transactions on Mathematical Software (TOMS)**, Volume 26 Issue 4

Full text available: pdf(139.60 KB) Additional Information: full citation, abstract, references, index terms, review

Many computationally intensive problems in engineering and science give rise to the solution of large, sparse, linear systems of equations. Fast and efficient methods for their soltion are very important because these systems usually occur in the innermost loop of the computational scheme. Parallelization is often necessary to achieve an acceptable level of performance. This paper presents the design, implementation, and interface of a library of Basic Linear Algebra Subroutines for sparse ...

Keywords: basic linear algebra subprograms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player